Qualify Exam. (Computer Architecture)

1. (20%) The following table lists the SPECINTC2006 benchmarks running on a CPU. Answer the following questions

Description	Name	Instruction Count x 10^9	CPI	Clock cycle time (seconds x 10^{-9})	Execution Time (seconds)	Reference Time (seconds)	SPECratio
Interpreted string processing	perl	2252	0.60	0.376	508	9770	19.2
Block-sorting compression	bzip2	2390	0.70	0.376	629	9650	15.4
GNU C compiler	gcc	794	1.20	0.376	358	8050	22.5
Combinatorial optimization	mcf	221	2.66	0.376	221	9120	41.2
Go game (AI)	go	1274	1.10	0.376	527	10490	19.9
Search gene sequence	hmmer	2616	0.60	0.376	590	9330	15.8
Chess game (AI)	sjeng	1948	0.80	0.376	586	12100	20.7
Quantum computer simulation	libquantum	659	0.44	0.376	109	20720	190.0
Video compression	h264avc	3793	0.50	0.376	713	22130	31.0
Discrete event simulation library	omnetpp	367	2.10	0.376	290	6250	21.5
Games/path finding	astar	1250	1.00	0.376	470	7020	14.9
XML parsing	xalancbmk	1045	0.70	0.376	275	6900	25.1
Geometric mean	–	–	–	–	–	–	25.7

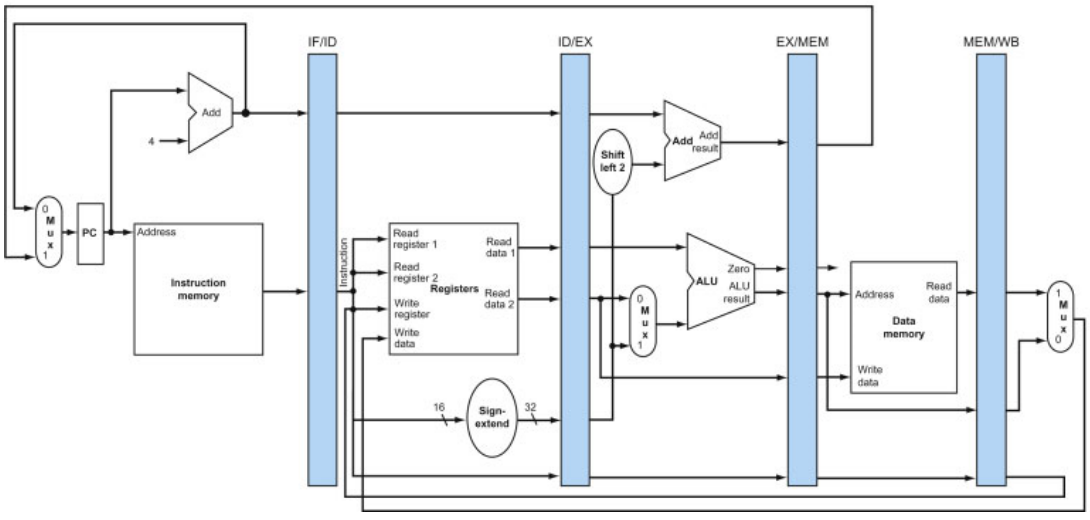
1.1 (5%) What is the running frequency of the CPU?

1.2 (5%) Explain how the execution time in the table is calculated. Hint: the execution time is calculated based on the information provided in the other columns of the table.

1.3 (5%) Explain how the SPECratio in the table is calculated.

1.4 (5%) Explain the possible reasons why some CPI values are smaller than 1.

2. (20%) The following problems refer to the pipelined datapath shown below which is a simplified version of MIPS pipeline.



The problems assume that, of all the instructions executed in a processor, the following fraction of these instructions have a particular type of RAW (Read-After-Write) data dependence.

Tab. 1

EX to 1st Only	MEM to 1st Only	EX to 2nd Only	MEM to 2nd Only	EX to 1st and MEM to 2nd	Other RAW Dependences
5%	20%	5%	10%	10%	10%

The type of RAW data dependence is identified by the stage that produces the result (EX or MEM) and the instruction that consumes the result (1st instruction that follows the one that produces the result, 2nd instruction that follows, or both). We assume that the register write is done in the first half of the clock cycle and the register reads are done in the second half of the clock cycle. Also, assume that the CPI of the processor is 1 if there are no data hazard.

Assume that the following latencies for individual pipeline stages. For the EX stage, latencies are given separately for a processor without forwarding and for a processor with different kinds of forwarding.

Tab. 2

IF	ID	EX (no FW)	EX (full FW)	EX (FW from EX/MEM only)	EX (FW from MEM/ WB only)	MEM	WB
150 ps	100 ps	120 ps	150 ps	140 ps	130 ps	120 ps	100 ps

2.1 (5%) Tab. 1 only shows the fraction for the 1st and 2nd next instructions that follow the instruction producing the results. Explain why it is not necessary to consider the 3rd next instruction that follows the instruction producing the results. For example, “EX to 3rd” and “MEM to 3rd” dependences are not counted.

2.2 (5%) If we do not use any forwarding, what fraction of cycles are we stalling due to data hazard? Hint: dependences to the 1st next instruction result in 2 stall cycles, and the stall is also 2 cycles if the dependence is to both 1st and 2nd next instruction. Dependences to only the 2nd next instruction result in one stall cycle. Calculate the CPI first, and then the fraction of stall cycles can be derived from the CPI.

2.3 (5%) If we use full forwarding (forward all results that can be forwarded), what fraction of cycles are we stalling due to data hazard? Hint: with full forwarding, the only RAW data dependences that cause stalls are those from the MEM stage of one instruction to the 1st next instruction. Even this dependences causes only one stall cycle.

2.4 (5%) For the given hazard probabilities and pipelined stage latencies, what is the speedup achieved by adding full forwarding to a pipeline that had no forwarding?

Hint: compute CPI without forwarding and with full forwarding. Then compute the clock cycle time in these two schemes. The execution time per instruction is the product of the above two values.

3. (20%) Cache Operation

Assume there is a small cache consisting of four one-word blocks. The following table shows the number of misses for the cache reference sequence of block addresses: 0, 8, 0, 6, 8 for a direct-mapped cache.

Address of memory block accessed	Hit or miss	Contents of cache blocks after references			
		Block 0	Block 1	Block 2	Block 3
0	miss	Memory[0]			
8	miss	Memory[8]			
0	miss	Memory[0]			
6	miss	Memory[0]		Memory[6]	
8	miss	Memory[8]		Memory[6]	

3.1 (10%) Draw a similar table to show the misses for the same cache access sequence for a two-way-set associative cache with the same cache size and block size (i.e., four one-word blocks). Assume that the replacement policy is least recently used (LRU). Among all the misses, how many misses are compulsory misses? How many misses are capacity misses? How many misses are conflict misses?

3.2 (10%) Repeat the above question for a full-associative cache of the same size.

4. (20%) Cache

4.1 **(5%)** How many bits in total (including the tag bits and valid bits) are required for a direct-map cache with 16K bytes of data and 16-byte blocks, assuming a 32-bit address and one valid bit for each cache block?

4.2 **(5%)** Repeat the above problem for a two-way set-associative cache.

4.3 **(5%)** Repeat the above problem for a fully associative cache.

4.4 **(5%)** Among three three cache placement policies of direct-map, set-associativity and fully-associativity, what placement policies are usually used for translation lookaside buffer (TLB) and why?

5. **(20%)** Answer the following questions

5.1 (5%) What is SIMD? Give brief explanation.

5.2 (5%) What is VLIW? Give brief explanation.

5.3 (5%) What is dynamic branch prediction? Explain the example of a 2-bit branch prediction scheme by drawing the state transition diagram.

5.4 (5%) What is static multiple issue? What is dynamic multiple issue? Compare the differences.